# 誤り訂正構文解析法 : 研究の現状と問題点(代数的コード理論および語の組合せ論)

AUTHOR(S):

田中, 栄一

CITATION:

田中, 栄一. 誤り訂正構文解析法 : 研究の現状と問題点(代数的コード理論および語の組合せ論). 数理解析研究所講究録 1992, 786: 1-14

ISSUE DATE:

1992-06

URL:

http://hdl.handle.net/2433/82593

数理解析研究所講究録
786巻 1992年 1-14



# 誤り訂正構文解析法 －研究の現状と問題点－

神戸大学工学部 田中栄一 (Eiichi Tanaka)

形式言語理論の立場から研究されている文脈自由言語の誤り訂正構文解析法の研究はある発展の段階に達していると思われるが、実際問題に応用しようとすると、計算量の問題に直面する。本文では研究の現状及び問題点を整理する。

## 1. 文の誤り訂正法

### 1.1 誤り訂正構文解析法

文法を$G=(N,T,P,S)$とする。ここで、N, T, P, Sはそれぞれ非終端記号の有限集合、終端記号の有限集合、生成規則の有限集合、文記号である。生成規制は$\alpha \rightarrow \beta$の形をした書き換え規則で、$\alpha \in V^*NV^*$、$\beta \in V^*$、$V=N \cup T$である。ここで$V^*$はVの要素から作られる全ての系列の集合である。$\alpha \in N$のとき、Gを文脈自由文法という。Sに生成規則を繰り返し適用して系列wが得られるとき、$S \overset{*}{\rightarrow} w$と書く。$S \overset{*}{\rightarrow} w$かつ$w \in T^*$のとき、

wはGが生成する文である。文法Gが定義する言語L(G)はGが生成する文の集合である。すなわち、$L(G)=\{w \mid S \overset{*}{\rightarrow} w,\ w \in T^*\}$。

構文解析法は文wが$w \in L(G)$かどうかを判定し、$w \in L(G)$ならwの構文解析木を作るアルゴリズムである。

2つの記号列を$A=a_1a_2\cdots a_m$, $B=b_1b_2\cdots b_n$とする。Aの文字が誤る(置換)、不用な文字が挿入する、Aの文字が脱落する、の3種類の誤りがあるとし、その重みを非負の値 p, q, r とする。系列Aから系列Bを得るのに必要な最小の重みでAからBへの距離D(A,B)を定義できる[1]。これを重みつきレーベンシュタイン距離と呼ぶ。Aから系列の集合L(G)への距離D(A,L(G))をAからL(G)の元への距離の最小値とする。

誤り訂正構文解析は文wに対してD(w,L(G))を計算し、D(w,x)=D(w,L(G)), $x \in L(G)$なるxがいかなる誤りによってwになったかを示す構文解析木を作るアルゴリズムである。

誤り訂正構文解析法には、文法に誤り操作を組み込んだものと構文解析過程で誤り訂正操作を行うものがある。前者は、正しい文を生成する文法Gの生成規則に誤りを発生する生成規則を付け加えた文法G'=(N, T, P', S)を作り、G'で文を構文解析して、誤り生成規則を最も少なく使った解析を誤り訂正構文解析とするものである[2]。この方法は一般にP'の規則が多く、計算手数が莫大になり実用的ではない。後者は構文

解析法の操作に誤り訂正操作を付け加えるもので、構文解析法に対応した誤り訂正構文解析法が存在する。すなわち、上昇形(CKY法)、下降形(Earley法)、混合形[3]の構文解析法を拡張した、上昇形[4]、[5]、下降形[6]、混合形[7]の誤り訂正構文解析法が存在する。いずれも入力の長さnに対して、$O(n^3)$の時間複雑さである。文脈自由言語及び文脈依存言語の誤り訂正構文解析法の現状については[8]に詳しいので、本文で必要になる部分の記述にとどめる。

## 1.2 構文解析法の音声認識への応用

音声認識に文脈自由言語の構文解析法を用いる試みがある[9]～[15]。形式言語の構文解析では入力は単語の列であるが、音声認識では入力は音韻ラティス或いは単語ラティスとして、音韻認識と構文処理を分離するものと、音韻認識と構文処理を融合しようとするものがある。また、単語や音韻の予測を行って、探索範囲の縮小を試みている。認識された音韻や単語の確からしさを表す確率情報を用いているので、CKY法を用いているもの[10]、[11]は、単語の置換誤り訂正法[16]に相当する操作が組み込まれていることになる。しかし、単語の挿入や脱落誤りがあれば、動作が停止してしまう。Earley法を用いた[9]も同様である。単語の挿入や脱落誤りの確率情報

が分かっておれば、[9]～[11]を訂正する操作を組み込むことは、誤り訂正構文解析法を参考にすれば、極めて容易である。拡張LR構文解析法を用いる方法[12]～[14]を単語の挿入や脱落誤りに対処できるようにすることについては、２.で言及する。

### 1.3 誤り訂正構文解析法の自然言語への適用

形式言語の構文解析法をそのまま、或いは修正して自然言語に適用するのはあまり問題がないように思われる。しかし、誤り訂正構文解析法を自然言語に適用すると莫大な計算時間と記憶量が必要になり、実用的ではない。筆者の経験では、自由言語の誤り訂正構文解析法を自然言語に適用すると、文法規則が数百程度になれば、処理過程で大量のデータを生成するため、1MB、1MIPSの計算機(ACOS-600S)では数語程度の文の解析しかできず、また1文の解析に数時間を必要とすることもあった。従って、音声認識で単語の挿入や脱落誤りを訂正するために、上昇形や下降形の誤り訂正構文解析法を用いると、計算時間と記憶量の問題に直面するのは明かである。

## ２．誤り訂正構文解析法に拡張されていない構文解析法

文脈自由言語の構文解析を少しでも高速にしようとする試

みが幾つかある。これらはすべて誤り訂正構文解析法に拡張できるはずである。

CKY法及びEarley法の時間複雑さは$O(n^3)$であるが、$O(n^{2.82})$で構文解析できるとする方法[17]がある。この方法は操作が複雑で、nが小さい場合はオーバーヘッドが大きく実用的でない。

またEarley法の改良[18]もある。次のような導出を予め調べておく。

del := {A | A∈V, A→ε} (εは空系列).

sub := {(A,B) | A,B∈V, A→B}.

first := {(A,B) | A,B∈V, A→Bη, η∈V*}.

いま、tree(i,j, A→αB・β)は入力$a_1a_2\cdots a_m$の一部分$a_{i+1}\cdots a_j$を生成規則 A→αBβのαBの部分で解析する解析木の集合とする。このとき、次の予測子を用いる。

(A→aα)∈P, a=$a_{j+1}$→・$a_{j+1}$∈tree(j,j+1,A→a・α)とする。

(A→Bα)∈P, tree(i,j+1,B→b・)≠φ

→tree(i,j+1,A→B・α)を作る。

導出に関する情報(del, sub, first)と上記の予測子を用いてEarley法を高速にすることができる。この方法もLyon法と同様な誤り訂正構文解析法に容易に拡張することができる。この方法がLyon法をどの程度改善したかについて知るには実

験的検討が必要である。

他の1つはLR言語[19]の構文解析法を文脈自由言語に適用できるようにした拡張LR構文解析法[20]と呼ばれるものがある。LR言語の構文解析はLR解析表を用いるもので、表は状態s、終端記号t、非終端記号nでの動作 act(s,t)及び goto(s,n)が記入されている。act(s,t)は次の入力を読み状態 q に遷移する (shift q)か、生成規則pの左辺の記号に置き換える(reduce p)か、受理するか、停止のいずれかの動作をする。goto(s,n)は次の遷移状態を示す。LR言語では動作が一意になるようにact(s,t)及びgoto(s,n)は高々1つの項目が書かれているだけであるが、文脈自由言語では複数の項目が記入されることがある。従って、バックトラックを避けるためには並列動作をすることになる。これを誤り訂正構文解析法に拡張しようとすると、状態sと終端記号$a_k$の対$(s,a_k)$に対して次の4つの動作をすればよい。

- act$(s,a_k)$を実行する。
- act$(s,b)$$(a_k \neq b)$を実行し置換の重みを加える。
- act$(s,a_{k+1})$を実行し挿入の重みを加える。
- act(s,b)が空でないbを探し実行する。bが脱落したとして脱落の重みを加える。状態がqに遷移したとする。続いてact$(p,a_k)$を実行する。

このようにして作った拡張LR誤り訂正構文解析法が実用に耐える高速性を持つかどうかは実験してみないとわからないように思われる。LR言語の誤り訂正法をPASCALの部分クラスに適用した経験では、上記の計算機で、10行程度のプログラムの誤り訂正に数秒かかることがあり、高速性を期待して自然言語の誤り訂正に拡張・援用することは無理であると思われたからである。計算時間はアルゴリズムとその実現法、文法の規模、辞書等に大きく依存するので実験的検証が必要である。

$O(n^2)$のプロセッサを使って$O(n)$時間で解析する並列構文解析法[21]もある。この方法も誤り訂正能力を持つように拡張できるだろう。

## ３．スタックを用いた誤り訂正構文解析法

これ迄スタックを用いた構文解析法を誤り訂正構文解析法に拡張することは無理だと考えられていた。時間複雑さが$O(c^n)$（cは常数）であるし、解析の途中結果を記憶していないのでいかにも誤り訂正には不向きであるように見える。しかし、文中の誤りが多いものは人でも理解できないから、誤りの数は少ないと仮定してよい。

入力語系列を$a_1 a_2 \cdots a_n$とする。$a_k$($k=1,2,\cdots,n$)は事実上品

詞である。解析の途中結果は項m:(k,A,B,x,y,z)で表し、k以外はスタックである。mは項番号、kは$a_k$に注目していることを示す。Aは$a_1$～$a_{k-1}$の解析に成功した規則の記号を記憶し、Bは$a_k$～$a_n$を導出する規則の記号を記憶する。xは操作の重み、yは誤り位置，zは誤った品詞を記憶する。項①m:(k,g,fh,x,y,z)に対して次の操作をする。

一致操作：$a_k$=fのとき、

②$m_1$:(k+1,gf,h,x,y,z)を作る。

置換誤り操作：f∈Tで、$a_k$≠fのとき、

③$m_2$:(k+1,gf,h,xp,yk,zf)を作る。

挿入誤り操作：f∈Tのとき、

④$m_3$:(k+1,g,fh,xq,yk,z_)を作る。

脱落誤り操作：f∈Tのとき、

⑤$m_4$:(k,gf,h,xr,yk,zf)を作る。

thを閾値として次の手順で解析する。また、非終端記号fを左辺に持つ生成規則がk個であるとし、それを$f_1 \rightarrow u_1$,$f_2 \rightarrow u_2$,…,$f_k \rightarrow u_k$とする。

begin

項 1:(1,φ,s,φ,φ,φ)

repeat

for m=1 to maxlist do

begin

if 項 m: (i,g,fh,$v_1$,$v_2$,$v_3$)が f∈ N then

begin

項mを消去して、

項 $m_1$ : (i, $gf_1$, $u_1h$, $v_1$, $v_2$, $v_3$)

項 $m_2$ : (i, $gf_2$, $u_2h$, $v_1$, $v_2$, $v_3$)

:

項 $m_k$ : (i, $gf_k$, $u_kh$, $v_1$, $v_2$, $v_3$)を作る。

end

else if 項m :(i, g, fh, $v_1$, $v_2$, $v_3$)かつ f∈ T then

begin

if $f=a_i$

then $m_1$ :(i+1, gf, h, $v_1$, $v_2$, $v_3$)を作る。

else $v_1+p \leqq th$ のとき、

$m_1$ :(i+1, gf, h, $v_1p$, $v_2$, $v_3f$)を作る。

(置換誤り訂正操作)

$v_2+q \leqq th$ のとき、

$m_2$ :(i+1, g, fh, $v_1$, $v_2i$, $v_3$_)を作る。

(挿入誤り訂正操作)

$v_2+r \leqq th$ のとき、

$m_3$ :(i,gf, h, $v_1$r, $v_2$i, $v_3$f)を作る。

（脱落誤り訂正操作

項mを消去する。

end

end

until（リスト中の全ての項の第３要素がφ）

end.

リストに残った項が成功した解析結果である。予備的な実験であるが、文中の誤りは高々１であるとして、これを英語に適したように修正して応用すると、上昇形、下降形の文脈自由言語の誤り訂正構文解析法に比べて非常に速い誤り訂正構文解析ができる[22]。まだ充分検討していないが、この結果は英文法を単純に文脈自由文法で近似することが適当でないことを示しているのかもしれない。

## ４．あとがき

現在知られている構文解析法及び誤り訂正構文解析法は共に時間複雑さが$O(n^3)$であるが、後者はある程度の規模の文法になると莫大な記憶量と計算時間が必要になり実用からはほ

ど遠い。$O(n^3)$と$O(n)$の差を埋めるためには意味情報を利用しなければならないことは云う迄もないが、自然言語を単純に文脈自由言語で近似することにも問題があるかもしれない。多くの場合、人は実時間で正しい文を認識する。更に、少ない誤りならほぼ実時間で正しく訂正して認識する。言語理論は言語認識のあり方をも説明するものであるべきだと思われるから、音声認識や自然言語処理の実際的な研究と共に、言語認識について理論的な関心が持たれてもよいように思う。

文献

[1] T. Okuda, E. Tanaka and T. Kasai : A garbled word correcting method by an extended metric, Ann. Joint Meeting of Elect. and Electro. Eng. of Tokai District, 18a-B-6, 1972, also A method for the correction of garbled words based on the Levenshtein metric, IEEE Trans. Computers, Vol.C-25, No.2, pp.172～178, 1976.

[2] A.V. Aho and T.G. Peterson : A minimum distance error-correcting parser for context-free languages, SIAM J. Computing, Vol.4, December 1972.

[3] S.L. Graham, M.A. Harrison and W.L. Ruzzo : An

improved context-free recognizer, ACM Trans. Program Lang. & Syst., Vol.2, pp.415～462, 1980.

[4] E. Tanaka and K.S. Fu : Error-correcting parsers for formal languages, Technical Report, Purdue Univ. 1976, also IEEE Trans. Computers, Vol.C-27, No.7, pp.605～617, 1978.

[5] 山崎、外村：文脈自由言語のbottom-up的最小誤り訂正について、情報処理、Vol.18, No.3, pp.781～788, 1977.

[6] G. Lyon : Syntax-directed least-errors analysis for context-free languages — a practical approach, Comm. ACM, Vol.17, pp.3～14, 1974.

[7] E. Tanaka : An improved error-correcting parser for a context-free language, Trans. IECE, Vol.E67, No.7, pp.379～385, July 1984.

[8] E. Tanaka : Parsing and error-correcting parsing for string grammars, Syntactic and Structural Pattern Recognition - Theory and Applications (Bunke and Sanfeliu eds), World Scientific, 1990.

[9] 中川聖一：文脈自由文法のフレーム同期型構文解析法による連続音声認識、電子通信学会論文誌D, Vol.J70-D, No.5, pp.907～916, 1987年5月。

[10]伊藤、牧野、城戸：機能語予測ＣＫＹ法による日本語文音声の統語処理、電子情報通信学会論文誌Ｄ－Ⅱ、Vol.J74-D-Ⅱ, No.9, pp.1147～1155, 1991年9月。

[11]H. Ney : Dynamic programming parsing for context-free grammars in continuous speech recognition, IEEE Trans. Signal Processing, Vol.39, No.2, pp.336～340, Feburuary 1991.

[12]M. Tomita : An efficient word lattice parsing algorithm for continuous speech recognition, Proc. ICASSP 86, pp.1569～1572, Tokyo 1986.

[13]斉藤、富田：ＬＲパーザによる誤りを含む文の認識、電子情報通信学会技術研究報告、SP88-28, 1988.

[14]北、川端、斉藤：ＨＭＭ音韻認識と拡張ＬＲ構文解析法を用いた連続音声認識、情報処理学会論文誌、Vol.31, No.3, pp.472～479, 1990年3月。

[15] 岡田美智男：アクティブチャート解析法に基づくOne-Passアルゴリズムの構文制御について、電子通信学会技術研究報告、SP90-24, 1990年。

[16]L.W. Fung and K.S. Fu : Maximum-likelihood syntactic decoding, IEEE Trans. Information Theory, Vol.IT-21, pp.423～430, 1975.

[17]L.G. Valiant : General contex-free recognition in less than cubic time, J. Computer and System Science, Vol.10, pp.308～315, 1975.

[18]H. Leiss : On Kilbury's modification of Earley's algorithm, ACM Trans. Programming Languages and Systems, Vol.12, No.4, pp.610～640, October 1990.

[19]D.E. Knuth : On the translation of language from left to right, Information and Control, Vol.8, pp.607～639, 1965.

[20]M. Tomita : An efficient context-free parsing algorithm for natural languages, Proc. 9th International Joint Conference on Artificial Intelligence, pp.756～764, 1985.

[21]峯、谷口、雨宮：一般の文脈自由文法に対する効率的な並列構文解析、情報処理学会論文誌、Vol.32, No.10, pp.1225～1237、1991年10月。

[22]田中栄一：英文高速解析システム(HISEAS)の考え方及び英文法、平成３年度電子情報通信学会信越支部大会、137, 1991年。

岩瀬、田中：文法を用いた英文の誤り訂正システム－HISEAS IV－、同上, 140, 1991年。